UNITED STATES PATENT AND TRADEMARK OFFICE

APPLICATION NO.	FILING DATE	FIRST NAMED INVENTOR	ATTORNEY DOCKET NO.	CONFIRMATION NO.
10/582,839	03/23/2007	Miten M. Marfatia	25086-11640	4990

7590   12/06/2010

Rajiv P Patel
Silicon Valley Center
Fenwick & West
801 California Street
Mountain View, CA 94041

EXAMINER
CHOWDHURY, ZIAUL A.

ART UNIT	PAPER NUMBER
2192	

MAIL DATE	DELIVERY MODE
12/06/2010	PAPER

Please find below and/or attached an Office communication concerning this application or proceeding.

The time period for reply, if any, is set in the attached communication.

PTOL-90A (Rev. 04/07)

-- The MAILING DATE of this communication appears on the cover sheet with the correspondence address --

Period for Reply

A SHORTENED STATUTORY PERIOD FOR REPLY IS SET TO EXPIRE *3* MONTH(S) OR THIRTY (30) DAYS, WHICHEVER IS LONGER, FROM THE MAILING DATE OF THIS COMMUNICATION.

- Extensions of time may be available under the provisions of 37 CFR 1.136(a). In no event, however, may a reply be timely filed after SIX (6) MONTHS from the mailing date of this communication.
- If NO period for reply is specified above, the maximum statutory period will apply and will expire SIX (6) MONTHS from the mailing date of this communication.
- Failure to reply within the set or extended period for reply will, by statute, cause the application to become ABANDONED (35 U.S.C. § 133). Any reply received by the Office later than three months after the mailing date of this communication, even if timely filed, may reduce any earned patent term adjustment. See 37 CFR 1.704(b).

Status

1)☒ Responsive to communication(s) filed on *20 September 2010*.

2a)☒ This action is **FINAL**.   2b)☐ This action is non-final.

3)☐ Since this application is in condition for allowance except for formal matters, prosecution as to the merits is closed in accordance with the practice under *Ex parte Quayle*, 1935 C.D. 11, 453 O.G. 213.

Disposition of Claims

4)☒ Claim(s) *1-18* is/are pending in the application.

　　4a) Of the above claim(s) _____ is/are withdrawn from consideration.

5)☐ Claim(s) _____ is/are allowed.

6)☒ Claim(s) *1-18* is/are rejected.

7)☐ Claim(s) _____ is/are objected to.

8)☐ Claim(s) _____ are subject to restriction and/or election requirement.

Application Papers

9)☐ The specification is objected to by the Examiner.

10)☐ The drawing(s) filed on _____ is/are: a)☐ accepted or b)☐ objected to by the Examiner.

　　Applicant may not request that any objection to the drawing(s) be held in abeyance. See 37 CFR 1.85(a).

　　Replacement drawing sheet(s) including the correction is required if the drawing(s) is objected to. See 37 CFR 1.121(d).

11)☐ The oath or declaration is objected to by the Examiner. Note the attached Office Action or form PTO-152.

Priority under 35 U.S.C. § 119

12)☐ Acknowledgment is made of a claim for foreign priority under 35 U.S.C. § 119(a)-(d) or (f).

　　a)☐ All   b)☐ Some * c)☐ None of:

　　　　1.☐ Certified copies of the priority documents have been received.

　　　　2.☐ Certified copies of the priority documents have been received in Application No. _____.

　　　　3.☐ Copies of the certified copies of the priority documents have been received in this National Stage application from the International Bureau (PCT Rule 17.2(a)).

　　* See the attached detailed Office action for a list of the certified copies not received.

Attachment(s)

1)☒ Notice of References Cited (PTO-892)

2)☐ Notice of Draftsperson's Patent Drawing Review (PTO-948)

3)☐ Information Disclosure Statement(s) (PTO/SB/08)
　　Paper No(s)/Mail Date _____.

4)☐ Interview Summary (PTO-413)
　　Paper No(s)/Mail Date. _____.

5)☐ Notice of Informal Patent Application

6)☐ Other: _____.

Detailed Action

Status of Claims

1.   Applicant's amended claims dated September 20th, 2010 responding to the June 23rd, 2010 Office Action provided in the rejection of claims 1-6.

2.   Claims 1-6 have been amended, and claims 7-18 have been added.

3.   Claims 1-18 are pending in the application, of which claims 1, 7, and 13 are in independent form.

4.   Applicant's amendments to the claims 1-6 do not over come the deficiencies stated in the previous Office Action for the said claims 1-6 rejected under 35 U.S.C. § 101. Therefore, rejections to the claim 1-6 under U.S.C. § 101 has been maintained. For further analysis please refer to the section providing rejections under 35 U.S.C. § 101 below.

Remarks

5.   Applicant's arguments with respect to claims rejection have been considered, but are moot in view of the new grounds of rejection.

–See Moore et al. (US Pub. No. 2004/0158820 A1), Lake et al. (US Pub. No 2003/0220920 A1), and Uchikawa et al (US Pat. No. 6,330,553 B1) – arts being made of record.

Specification

6.   The specification is objected to as failing to provide proper antecedent basis for the claimed subject matter. See 37 CFR 1.75(d)(1) and MPEP § 608.01(o).

        Claims 1, 7 and 13-18 recite phrase "non-transitory computer readable storage medium"; however, the originally filled disclosure fails to provide any

information regarding this said phrase "non-transitory computer readable storage medium". Therefore, the applicant is suggested to make appropriate amendment to the disclosure to present clear support or antecedent basis for the terms appeared in the said claims; however, no new matter should be introduced.

Claim Rejections - 35 USC § 101

35 U.S.C. 101 reads as follows:

> Whoever invents or discovers any new and useful process, machine, manufacture, or composition of matter, or any new and useful improvement thereof, may obtain a patent therefor, subject to the conditions and requirements of this title.

7.   Claims 1-6 and 13-18 are rejected under 35 U.S.C. § 101 because the claimed invention is directed to non-statutory subject matter.

Per claim 1, claim 1 has been previously rejected under 35 U.S.C. § 101 because the claimed limitations do not fall within any of the four statutory categories of the invention; however, in order to over come the previous rejection under 35 U.S.C. 101 applicants have added the phrase "the custom knowledge base comprises a relational database comprising source and target code patterns and attributes and residing on a non-transitory computer-readable storage medium". Even though "a non-transitory computer readable storage medium" is a statutory subject matter; however, as indicated in the previous section of this action claimed limitation has not been supported by the originally filed disclosure. Furthermore, applicants' provided written response has not identified any section from the specification which may support the amended claim's limitations. Accordingly, for the compact prosecution purpose examiner will consider these terms as statutory upon perfection and appropriate amendment to the specification; otherwise, examiner will reasonably interpret the functionality of this apparatus as an implementation of software alone, Software and computer program *per se* do

not fall within any category of patent-eligible subject matter. Claim 1 will therefore deem to be ineligible for patent protection. *See* MPEP § 2106.

Per claim 13, the newly added claim 13 recites "A non-statutory computer-readable storage medium encoded .... on a target platform". However, originally filed disclosure does not provide any information regarding this medium and/or a computer readable storage where the encoded software instruction or code for implementing system and/or method recorded thereon. As indicated above, for the compact prosecution purpose examiner will consider these terms as statutory upon perfection and appropriate amendment to the specification; otherwise, product claim without a tangible medium would be reasonably interpreted as entirely software. Software and computer program *per se* do not fall within any category of patent-eligible subject matter. Therefore, it is clear that claim 13 does not fall within any of the categories of patentable subject matter set forth in 35 U.S.C. § 101, and such claim is therefore ineligible for patent protection. See MPEP § 2106.

Claims 14-18 depend from claim 13, and therefore, can not overcome the deficiency stated above; accordingly claims 14-18 are also being rejected as non-statutory subject matter.

Claim Rejections - 35 USC § 103

The following is a quotation of 35 U.S.C. 103(a) which forms the basis for all obviousness rejections set forth in this Office action:

> (a) A patent may not be obtained though the invention is not identically disclosed or described as set forth in section 102 of this title, if the differences between the subject matter sought to be patented and the prior art are such that the subject matter as a whole would have been obvious at the time the invention was made to a person having ordinary skill in the art to which said subject matter pertains. Patentability shall not be negatived by the manner in which the invention was made.

8.   Claims 1-4, 7, 8, 11-14, 17 and 18 are rejected under 35 U.S.C. 103(a)
as being unpatentable over Pastukha et al. (US Pat. No 5,678,044 –IDS of
record) in view of Moore et al. (US Pub. No 2004/0158820 A1 –herein after
Moore), and Lake et al. (US Pub. No. 2003/0220920 –herein after Lake).

Per claim 1:

Pastilha discloses

An apparatus for converting migration and conversion of software code of a
        source application on a from any source platform into software code of a
        target application on a target platform (Col 2:45-48 –The system
        (apparatus) and method of the present invention achieves these and
        other objects by providing a computerized tool (apparatus) for
        <u>automating the processes necessary to successfully migrate a software</u>
        <u>system from one environment to another</u> –emphasis added) the
        apparatus comprising:

an inputting means for accepting the source code of the source application to
        analyze business logic of the source application (Col 1:32-38 –Rehosting,
        however, is becoming more often a viable strategy for moving legacy
        applications from a mainframe (source) to a mid-range (target) platform.
        Rehosting is the physical movement of one or more <u>business applications</u>
        from one computing environment, such as mainframe, to another
        computing environment, such as a mid-range computing platform; Col
        3:45-49 –The invention consists of components which automatically
        gather the information used <u>to evaluate candidate applications</u> (by an
        input means), which automatically convert software systems from their
        source operating environment to a new and different targeted
        environment),
        obtaining a definitions of a target back-end system (Col 5:60-62 –
        heuristics 190 determine whether to rehost application software 130

based on the results of discovery 140 <u>in view of the designated target</u>

<u>computing environment</u>), (2) *existing test scripts to facilitate quality*

control of generated software code for the target application (Col 10:11-14

–Following the <u>analysis of application-specific data</u> gathered by discovery

140, a <u>candidate software application</u> (or system) 130 will be judged as

either suitable for rehosting or not suitable), (3) *source code entry points*

to business processes (Col 1:34-38 –Rehosting is the physical movement

of one or more <u>business applications</u> from one computing environment,

such as mainframe, to another computing environment, such as a mid-

range computing platform), (4) *target environment specification including*

the target platform (Col 5:60-62 –heuristics 190 determine whether to

rehost application software 130 based on the results of discovery 140 <u>in</u>

<u>view of the designated target computing environment</u>), *languages to be*

used (Col 19:32-37 –a computer processor having a migration utility

rehosting at least one job, at least one proc and at least one file of said

software system to said target computing environment and converting job

control language of said source computing environment into an interim

rehosting language), *target database* (Refer to FIG 7, 480; and associated

text), *coding standards* (Col 10:11-14 –Following the analysis of

application-specific data gathered by discovery 140, a candidate software

application (or system) 130 will be judged as either suitable for rehosting

or not suitable), target architecture and framework (Col 6:7-11 –estimate

the resources required on a target mid-range platform if the system is

migrated; estimate the complexity of the programs and jobs by analyzing

job history (e.g.: CPU time vs. overall elapsed time) and program

attributes (e.g.: language, size), manage the system migration effort, etc),

third party components (Col 5:17-21 –some of the support features

included as run-time services are Generation Data Group (GDG)

administration, program and/or job submission and execution based on

system return codes, file disposition processing and other production

system monitoring activities), (6) *existing applications to be plugged with*

the target application (Col 3:16-21 -A run-time services utility provides

operational system support for the rehosted software system now within

the target computing environment similar to operational system support

previously available in the source computing environment based on the

assessment and description), and (7) *sample code for the target*

application (Col 3:29-31 –A run-time services utility residing on the

computer processor and generates execution scripts from the interim

rehosting language);

an analyzing means for analyzing provided source schemes to create target

schemes (Col 2:66-67, Col 3:1-3 –The discovery utility (knowledge

Engine) is also suited to <u>analyze the predetermined information to assess</u>

<u>the feasibility of rehosting the candidate software application</u> from the

source computing environment to the target computing environment),

analyzing the business logic of the source application to create workflow

diagrams that represent processes of the source application (Col 1:34-38 –

Rehosting is the physical movement of one or more <u>business applications</u>

from one computing environment, such as mainframe, to another

computing environment, such as a mid-range computing platform),

identifying code segments of the source application (Col 5:50-52 -Source

code analyzer 160 interacts with data selector 150 and RDBMS 155 to

analyze the source code of each application software 130), and *analyzing*

the target environment to generate a target architecture and associated

technology (Col 5:60-67 -heuristics 190 determine whether to rehost

application software 130 <u>based on the results of discovery</u> 140 in view of

the designated <u>target computing environment</u>. Such embodiment of the

present invention would <u>perform a rules-based analysis</u> of the reports

generated by report generator 180 to recommend the class of machine to

use, and <u>to suggested configuration, in the target computing</u>

<u>environment</u>);

a setting up means for generating a custom knowledge base for the software

code conversion that is responsive to no existing knowledge base for

particular migration existing (Col 5:50-56 -Source code analyzer 160

interacts with data selector 150 and RDBMS 155 to analyze the source

code of each application software 130. <u>The results of such analysis is</u>

<u>also stored in RDBMS</u> (generating a knowledgebase)155. Similarly, JCL

source analyzer 170 interacts with RDBMS 155 to analyze the JCL

source and <u>store the results of such analysis in RDBMS</u> (generating a

knowledgebase)155), *wherein the custom knowledge base comprises a*

relational database comprising source and target code patterns and

attributes and residing on a non-transitory computer-readable storage

medium (Col 5:43-50 -Discovery portion 140 of the present invention

comprises requesting portion 145 which requests certain information

from source operating system 110, <u>any database(s) 125</u> (knowledge base)

and application software 130. Requesting portion 145 provides the

results of its inquiries to data selector 150 for <u>extraction of certain</u>

<u>predetermined pieces of data therefrom</u>. These pieces of <u>data are stored</u>

<u>in relational database</u> management system ("RDBMS") 155);

a processing means for conversion of source code into a format of the target

environment specification (Col 3:56-62 –This is achieved by the present

invention by gathering information about the software application

candidate for rehosting (including the source computing environment in

which it operates), converting data files and related job control

statements used on virtually any source host to the <u>corresponding</u>

<u>formats required by a new targeted host processing environment</u>, Col

10:28-31 –Migration 400 automatically <u>converts</u> all files and data

elements <u>from the formats used on source operating system 110 to</u> the

format requirements of <u>target operating system 450</u>), *the knowledge engine remains coupled to the custom knowledge base during the plurality of iterations for conversion of the source code into the format of the target environment specification* (Col 3:56-62 –This is achieved by the present invention by gathering information about the software application candidate for rehosting (including the source computing environment in which it operates), converting data files and related job control statements used on virtually any source host to the corresponding formats required by a new targeted host processing environment, Col 10:28-31 –Migration 400 automatically converts all files and data elements from the formats used on source operating system 110 to the format requirements of target operating system 450), *the knowledge base is updated to include additional which leads structured information of the source platform and the source application with respect to the target platform and the target environment specification after each iteration to cause the knowledge engine to enhance source code conversion in subsequent iterations* (Col 2:66-67, Col 3:1-5 - The discovery utility (knowledge Engine) is also suited to <u>analyze the predetermined information to assess the feasibility of rehosting the candidate software application</u> from the source computing environment to the target computing environment. <u>A relational database management system receives and stores the predetermined information and the results of the analysis performed by the discovery utility</u> (Thus, relational database saves information for next update on source application according to the information stored and updated in the relational database) –emphasis added); and

a documenting means for generation of a report comprising a portion of the source code of the source application that is not converted automatically for manual conversion (Col 3:35-40 -The system takes record descriptors

(generation of reports) of data in the source environment that needs to be migrated and generates conversion programs for the data that get compiled. The compiled programs are then executed against the source data, producing output files that can be ported to the target environment; Col 9:43-47 –Based on the reports generated, a recommendation may be made by the present invention that the data be kept on a mainframe as a data repository while the application itself be ported to a mid-range environment; Col 9:64-67 –A further illustration might be a case where the present invention "discovers" that the programs in the application to be rehosted consist of a mix of program languages, some of which cannot reasonably be converted by any means other than re-writing);

Pastilha sufficiently discloses a system and method for automatically re-hosting (migrating) software system from a source computing environment to target computing environment, wherein the said system and method provide an utility component to assess both the source and target computing environment based on the information gathered from a relational knowledge base regarding said source and target environment, extracted from application software and predetermined pieces of data, and further analyze the assessment, and generates a description about both source and target computing environments. A migration utility propagate re-hosting or migration process according to the generated assessment and description (—See Pastilha Abstract, and Col 2-6), but Pastilha does not explicitly discloses— obtaining User Interface (UI) details of the source application receiving a validation scheme of a source front-end interface, wherein the source code is passed through a knowledge engine for a plurality of iterations;

however, Moore discloses—

obtaining User Interface (UI) details of the source application receiving a

>   *validation scheme of a source front-end interface* (Refer to FIG. 4, 168,
>   and associated text);

Thus it would have been obvious to one ordinary skill in the art at time the
invention was made to incorporate Moore's technique into Pastilha's invention
because the combined inventive system and method would provide an enhance
capability to automatically generate abstract intermediate representation of UI
information (front end), wherein the abstract representation would be used to
generate corresponding components in a new application shell in target
language or architecture (See Moore Paragraph [0039]), but Pastilha modified
by Moore does disclose: a plurality of iterations;

however, Lake discloses—

>   wherein the source code is passed through a knowledge engine for a

plurality of iterations (Paragraph [0064] -The general methods of FIGS. 5A and
5B may be applied over multiple iterations in order to fully update a database,
each iteration using a different implementation of the methods directed at
different fields of the databases).

It would have been obvious to one ordinary skill in art at the time the invention
was made to use Lake's teaching in Pastilha modified by Moore's invention
because this inventive technique would enhance the discovery utility of
Pastilha to import the source and target environment information and extract
attributes from subjected application in order to implement the migration
process, and each iteration of the discovery utility may analyze attributes that
were generated through an earlier iteration, and update the database with the
new attributes (—See Lake Paragraph [0064]).

Per claim 2:

Pastilha discloses

>   *wherein an existing knowledge base is provided for understanding the*

source application, and the source platform, the target environment specification,

and the target platform (Col 2:61-67, Col 3:1-3 - A computer processor having a discovery utility is used to request predetermined information from the source computing environment including the operating system, any databases in the source computing environment, as well as from the candidate software application. The discovery utility is also suited to analyze the predetermined information to assess the feasibility of rehosting the candidate software application from the source computing environment to the target computing environment).

Per claim 13:

Pastilha discloses

   wherein the processing means is further configured a to extract the business logic and database schema of the source application systematically and logically and to convert them into a format specified for the target application (Col 5:43-50 -Discovery portion 140 of the present invention comprises requesting portion 145 which requests certain information from source operating system 110, <u>any database(s) 125</u> (knowledge base) and application software 130. Requesting portion 145 provides the results of its inquiries to data selector 150 for <u>extraction of certain predetermined pieces of data therefrom</u>. These pieces of <u>data are stored in relational database</u> management system ("RDBMS") 155; Col 134-38 –Rehosting is the physical movement of one or more business applications from one computing environment, such as mainframe, to another computing environment, such as a mid-range computing platform).

Per claim 4:

Pastilha discloses

   wherein the processing means is further configured to dynamically hatch new patterns to be used to convert the source code into the format of the target

environment specification (Col 14:61-65 - if the source operating system from which the software is migrated does not require iterative file set handling, then the present <u>invention will create an equivalent of such iterative file set(s) for improved functionality</u> once the migrated software has been rehosted).

Per claim 7:

Pastilha discloses

A method for converting software code of a source application on a source platform into software code of a target application on a target platform (Col 2:45-48 –The system (apparatus) and method of the present invention achieves these and other objects by providing a computerized tool (apparatus) for <u>automating the processes necessary to successfully migrate a software system from one environment to another</u> –emphasis added), the method comprising:
accepting the source code of the source application to analyze business logic of

> *the source application* (Col 1:32-38 –Rehosting, however, is becoming more often a viable strategy for moving legacy applications from a mainframe (source) to a mid-range (target) platform. Rehosting is the physical movement of one or more business applications from one computing environment, such as mainframe, to another computing environment, such as a mid-range computing platform; Col 3:45-49 –The invention consists of components which automatically gather the information used <u>to evaluate candidate applications</u> (by an input means), which automatically convert software systems from their source operating environment to a new and different targeted environment);
>
> receiving a validation scheme of a source front-end interface (Col 4:62-67 –The discovery portion of the present invention supports a process which is an information-gathering function. Its purpose in a rehosting effort is to ascertain and <u>document</u> all <u>pertinent information</u>

about an <u>application under consideration</u> for rehosting, including the

environment in which it presently operates);

obtaining (1) a definition of a target back-end system (Col 5:60-62 –heuristics

190 determine whether to rehost application software 130 based on the

results of discovery 140 <u>in view of the designated target computing</u>

<u>environment</u>), (2) *existing test scripts to facilitate quality control of*

generated software code for the target application (Col 10:11-14 –

Following the analysis of application-specific data gathered by discovery

140, a candidate software application (or system) 130 will be judged as

either suitable for rehosting or not suitable), (3) *source code entry points*

to business processes (Col 1:34-38 –Rehosting is the physical movement

of one or more business applications from one computing environment,

such as mainframe, to another computing environment, such as a mid-

range computing platform), (4) *target environment specification including*

the target platform (Col 5:60-62 –heuristics 190 determine whether to

rehost application software 130 based on the results of discovery 140 <u>in</u>

<u>view of the designated target computing environment</u>), *languages to be*

used (Col 19:32-37 –a computer processor having a migration utility

rehosting at least one job, at least one proc and at least one file of said

software system to said target computing environment and converting job

control language of said source computing environment into an interim

rehosting language), *target database* (Refer to FIG 7, 480; and associated

text), *coding standards* (Col 10:11-14 –Following the analysis of

application-specific data gathered by discovery 140, a candidate software

application (or system) 130 will be judged as either suitable for rehosting

or not suitable), *target architecture and framework* (Col 6:7-11 –estimate

the resources required on a target mid-range platform if the system is

migrated; estimate the complexity of the programs and jobs by analyzing

job history (e.g.: CPU time vs. overall elapsed time) and program

attributes (e.g.: language, size), manage the system migration effort, etc),
(5) *third party components* (Col 5:17-21 –some of the support features
included as run-time services are Generation Data Group (GDG)
administration, program and/or job submission and execution based on
system return codes, file disposition processing and other production
system monitoring activities), (6) *existing applications to be plugged with
the target*
application (Col 3:16-21 -A run-time services utility provides operational
system support for the rehosted software system now within the target
computing environment similar to operational system support previously
available in the source computing environment based on the assessment
and description), and (7) *sample code for the target application* (Col 3:29-
31 –A run-time services utility residing on the computer processor and
generates execution scripts from the interim rehosting language);

analyzing provided source schemes to create target schemes (Col 2:66-67, Col
3:1-3 –The discovery utility (knowledge Engine) is also suited to <u>analyze
the predetermined information to assess the feasibility of rehosting the
candidate software application</u> from the source computing environment
to the target computing environment);

*analyzing the business logic of the source application to create workflow
diagrams that represent processes of the source application* (Col 1:34-38
–Rehosting is the physical movement of one or more business
applications from one computing environment, such as mainframe, to
another computing environment, such as a mid-range computing
platform);

identifying code segments of the source application (Col 5:50-52 -Source code
analyzer 160 interacts with data selector 150 and RDBMS 155 to analyze
the source code of each application software 130);

analyzing the target environment to generate a target architecture and

associated technology (Col 5:60-67 -heuristics 190 determine whether to
rehost application software 130 <u>based on the results of discovery</u> 140 in
view of the designated <u>target computing environment</u>. Such embodiment
of the present invention would <u>perform a rules-based analysis</u> of the
reports generated by report generator 180 to recommend the class of
machine to use, and <u>to suggested configuration, in the target computing
environment</u>);

*responsive to no existing knowledge base for the software code conversion
existing, generating a custom knowledge base for the software code
conversion* (Col 5:50-56 -Source code analyzer 160 interacts with data
selector 150 and RDBMS 155 to analyze the source code of each
application software 130. <u>The results of such analysis is also stored in
RDBMS</u> (generating a knowledgebase)155. Similarly, JCL source analyzer
170 interacts with RDBMS 155 to analyze the JCL source and <u>store the
results of such analysis in RDBMS</u> (generating a knowledgebase)155),
wherein the custom knowledge base comprises a relational database
comprising source and target code patterns and attributes and residing
on a non-transitory computer-readable storage medium (Col 5:43-50 -
Discovery portion 140 of the present invention comprises requesting
portion 145 which requests certain information from source operating
system 110, <u>any database(s) 125</u> (knowledge base) and application
software 130. Requesting portion 145 provides the results of its inquiries
to data selector 150 for <u>extraction of certain predetermined pieces of data
therefrom</u>. These pieces of <u>data are stored in relational database</u>
management system ("RDBMS") 155);

*converting the source code into a format of the target environment
specification* (Col 3:56-62 –This is achieved by the present invention by
gathering information about the software application candidate for
rehosting (including the source computing environment in which it

operates), converting data files and related job control statements used on virtually any source host to the <u>corresponding formats required by a new targeted host processing environment</u>, Col 10:28-31 –Migration 400 automatically <u>converts</u> all files and data elements <u>from the formats used on source operating system 110 to</u> the format requirements of <u>target operating system 450</u>), *the knowledge engine remains coupled to the custom knowledge base during the plurality of iterations for conversion of the source code into the format of the target environment specification* (Col 3:56-62 –This is achieved by the present invention by gathering information about the software application candidate for rehosting (including the source computing environment in which it operates), converting data files and related job control statements used on virtually any source host to the corresponding formats required by a new targeted host processing environment, Col 10:28-31 –Migration 400 automatically converts all files and data elements from the formats used on source operating system 110 to the format requirements of target operating system 450), *the custom knowledge base is updated to include additional structured information of the source platform and the source application with respect to the target platform and the target environment specification after each iteration to cause the knowledge engine to enhance source code conversion in subsequent iterations* (Col 2:66-67, Col 3:1-5 - The discovery utility (knowledge Engine) is also suited to <u>analyze the predetermined information to assess the feasibility of rehosting the candidate software application</u> from the source computing environment to the target computing environment. <u>A relational database management system receives and stores the predetermined information and the results of the analysis performed by the discovery utility</u> (Thus, relational database saves information for next update on source application

according to the information stored and updated in the relational
database) –emphasis added); and

*generating a report comprising a portion of the source code of the source
application that is not converted automatically for manual conversion (Col
3:35-40 -The system takes record descriptors* (generation of reports) of
data in the source environment that needs to be migrated and generates
conversion programs for the data that get compiled. The compiled
programs are then executed against the source data, producing output
files that can be ported to the target environment; Col 9:43-47 –Based on
the reports generated, a recommendation may be made by the present
invention that the data be kept on a mainframe as a data repository
while the application itself be ported to a mid-range environment; Col
9:64-67 –A further illustration might be a case where the present
invention "discovers" that the programs in the application to be rehosted
consist of a mix of program languages, some of which cannot reasonably
be converted by any means other than re-writing);

Pastilha sufficiently discloses a system and method for automatically re-
hosting (migrating) software system from a source computing environment to
target computing environment, wherein the said system and method provide an
utility component to assess both the source and target computing environment
based on the information gathered from a relational knowledge base regarding
said source and target environment, extracted from application software and
predetermined pieces of data, and further analyze the assessment, and
generates a description about both source and target computing environments.
A migration utility propagate re-hosting or migration process according to the
generated assessment and description (—See Pastilha Abstract, and Col 2-6),
but Pastilha does not explicitly discloses— obtaining User Interface (UI) details
of the source application receiving a validation scheme of a source front-end

interface, wherein the source code is passed through a knowledge engine for a plurality of iterations;

however, Moore discloses—

obtaining User Interface (UI) details of the source application receiving a validation scheme of a source front-end interface (Refer to FIG. 4, 168, and associated text);

Thus it would have been obvious to one ordinary skill in the art at time the invention was made to incorporate Moore's technique into Pastilha's invention because the combined inventive system and method would provide an enhance capability to automatically generate abstract intermediate representation of UI information (front end), wherein the abstract representation would be used to generate corresponding components in a new application shell in target language or architecture (See Moore Paragraph [0039]), but Pastilha modified by Moore does disclose: a plurality of iterations;

however, Lake discloses—

wherein the source code is passed through a knowledge engine for a plurality of iterations (Paragraph [0064] -The general methods of FIGS. 5A and 5B may be applied over multiple iterations in order to <u>fully update a database</u>, each <u>iteration using a different implementation</u> of the methods directed at different fields of the databases).

It would have been obvious to one ordinary skill in art at the time the invention was made to use Lake's teaching in Pastilha modified by Moore's invention because this inventive technique would enhance the discovery utility of Pastilha to import the source and target environment information and extract attributes from subjected application in order to implement the migration process, and each iteration of the discovery utility may analyze attributes that were generated through an earlier iteration, and update the database with the new attributes (—See Lake Paragraph [0064]).

Per claim 8:

Pastilha discloses

dynamically hatch new patterns to be used to convert the source code into the format of the target environment specification (Col 14:61-65 - if the source operating system from which the software is migrated does not require iterative file set handling, then the present <u>invention will create an equivalent of such iterative file set(s) for improved functionality</u> once the migrated software has been rehosted).

Per claim 11:

Pastilha discloses

wherein an existing knowledge base is provided for understanding the source application, the source platform, the target environment specification, and the target platform (Col 2:61-67, Col 3:1-3 - A computer processor having a discovery utility is used to request predetermined information from the source computing environment including the operating system, any databases in the source computing environment, as well as from the candidate software application. The discovery utility is also suited to analyze the predetermined information to assess the feasibility of rehosting the candidate software application from the source computing environment to the target computing environment).

Per claim 12:

Pastilha discloses

extracting the business logic and database schema of the source application systematically and logically (Col 5:43-50 -Discovery portion 140 of the present invention comprises requesting portion 145 which requests certain information from source operating system 110, <u>any database(s) 125</u> (knowledge base) and application software 130. Requesting portion 145 provides the

results of its inquiries to data selector 150 for <u>extraction of certain</u>
<u>predetermined pieces of data therefrom</u>. These pieces of <u>data are stored in</u>
<u>relational database</u> management system ("RDBMS") 155; Col 134-38 –
Rehosting is the physical movement of one or more business applications from
one computing environment, such as mainframe, to another computing
environment, such as a mid-range computing platform); and

converting the extracted business logic and database schema of the source
application into a format specified for the target application (Col 5:43-50 -
Discovery portion 140 of the present invention comprises requesting portion
145 which requests certain information from source operating system 110, <u>any</u>
<u>database(s) 125</u> (knowledge base) and application software 130. Requesting
portion 145 provides the results of its inquiries to data selector 150 for
<u>extraction of certain predetermined pieces of data therefrom</u>. These pieces of
<u>data are stored in relational database</u> management system ("RDBMS") 155; Col
134-38 –Rehosting is the physical movement of one or more business
applications from one computing environment, such as mainframe, to another
computing environment, such as a mid-range computing platform).

Per claim 13:

A non-transitory computer-readable storage medium encoded with
    executable computer program code for converting software code of a
    source application on a source platform into software code of a target
    application on a target platform (Col 2:45-48 –The system (apparatus)
    and method of the present invention achieves these and other objects by
    providing a computerized tool (apparatus) for <u>automating the processes</u>
    <u>necessary to successfully migrate a software system from one</u>
    <u>environment to another</u> –emphasis added), the computer program code
    comprising program code for:
accepting the source code of the source application to analyze business logic of

the source application (Col 1:32-38 –Rehosting, however, is becoming more often a viable strategy for moving legacy applications from a mainframe (source) to a mid-range (target) platform. Rehosting is the physical movement of one or more business applications from one computing environment, such as mainframe, to another computing environment, such as a mid-range computing platform; Col 3:45-49 –The invention consists of components which automatically gather the information used to evaluate candidate applications (by an input means), which automatically convert software systems from their source operating environment to a new and different targeted environment);

receiving a validation scheme of a source front-end interface (Col 4:62-67 –The discovery portion of the present invention supports a process which is an information-gathering function. Its purpose in a rehosting effort is to ascertain and document all pertinent information about an application under consideration for rehosting, including the environment in which it presently operates);

obtaining (1) a definition of a target back-end system (Col 5:60-62 –heuristics 190 determine whether to rehost application software 130 based on the results of discovery 140 in view of the designated target computing environment), (2) *existing test scripts to facilitate quality control of generated software code for the target application* (Col 10:11-14 – Following the analysis of application-specific data gathered by discovery 140, a candidate software application (or system) 130 will be judged as either suitable for rehosting or not suitable), (3) *source code entry points to business processes* (Col 1:34-38 –Rehosting is the physical movement of one or more business applications from one computing environment, such as mainframe, to another computing environment, such as a mid-range computing platform), (4) *target environment specification including the target platform* (Col 5:60-62 –heuristics 190 determine whether to

rehost application software 130 based on the results of discovery 140 <u>in view of the designated target computing environment</u>), *languages to be used* (Col 19:32-37 –a computer processor having a migration utility rehosting at least one job, at least one proc and at least one file of said software system to said target computing environment and converting job control language of said source computing environment into an interim rehosting language), *target database* (Refer to FIG 7, 480; and associated text), *coding standards* (Col 10:11-14 –Following the analysis of application-specific data gathered by discovery 140, a candidate software application (or system) 130 will be judged as either suitable for rehosting or not suitable), *target architecture and framework* (Col 6:7-11 –estimate the resources required on a target mid-range platform if the system is migrated; estimate the complexity of the programs and jobs by analyzing job history (e.g.: CPU time vs. overall elapsed time) and program attributes (e.g.: language, size), manage the system migration effort, etc), (5) *third party components* (Col 5:17-21 –some of the support features included as run-time services are Generation Data Group (GDG) administration, program and/or job submission and execution based on system return codes, file disposition processing and other production system monitoring activities), (6) *existing applications to be plugged with the target application* (Col 3:16-21 -A run-time services utility provides operational system support for the rehosted software system now within the target computing environment similar to operational system support previously available in the source computing environment based on the assessment and description), and (7) *sample code for the target application* (Col 3:29-31 –A run-time services utility residing on the computer processor and generates execution scripts from the interim rehosting language);

analyzing provided source schemes to create target schemes (Col 2:66-67, Col

3:1-3 –The discovery utility (knowledge Engine) is also suited to <u>analyze</u> <u>the predetermined information to assess the feasibility of rehosting the</u> <u>candidate software application</u> from the source computing environment to the target computing environment);

analyzing the business logic of the source application to create workflow diagrams that represent processes of the source application (Col 1:34-38 –Rehosting is the physical movement of one or more business applications from one computing environment, such as mainframe, to another computing environment, such as a mid-range computing platform);

identifying code segments of the source application (Col 5:50-52 -Source code analyzer 160 interacts with data selector 150 and RDBMS 155 to analyze the source code of each application software 130);

analyzing the target environment to generate a target architecture and associated technology (Col 5:60-67 -heuristics 190 determine whether to rehost application software 130 <u>based on the results of discovery</u> 140 in view of the designated <u>target computing environment</u>. Such embodiment of the present invention would <u>perform a rules-based analysis</u> of the reports generated by report generator 180 to recommend the class of machine to use, and <u>to suggested configuration, in the target computing environment</u>);

responsive to no existing knowledge base for the software code conversion existing, generating a custom knowledge base for the software code conversion (Col 5:50-56 -Source code analyzer 160 interacts with data selector 150 and RDBMS 155 to analyze the source code of each application software 130. <u>The results of such analysis is also stored in RDBMS</u> (generating a knowledgebase)155. Similarly, JCL source analyzer 170 interacts with RDBMS 155 to analyze the JCL source and <u>store the results of such analysis in RDBMS</u> (generating a knowledgebase)155),

wherein the custom knowledge base comprises a relational database
comprising source and target code patterns and attributes and residing
on a non-transitory computer-readable storage medium (Col 5:43-50 -
Discovery portion 140 of the present invention comprises requesting
portion 145 which requests certain information from source operating
system 110, any database(s) 125 (knowledge base) and application
software 130. Requesting portion 145 provides the results of its inquiries
to data selector 150 for extraction of certain predetermined pieces of data
therefrom. These pieces of data are stored in relational database
management system ("RDBMS") 155);

converting the source code into a format of the target environment
   specification (Col 3:56-62 –This is achieved by the present invention by
   gathering information about the software application candidate for
   rehosting (including the source computing environment in which it
   operates), converting data files and related job control statements used
   on virtually any source host to the corresponding formats required by a
   new targeted host processing environment, Col 10:28-31 –Migration 400
   automatically converts all files and data elements from the formats used
   on source operating system 110 to the format requirements of target
   operating system 450), *the knowledge engine remains coupled to the*
   custom knowledge base during the plurality of iterations for conversion of
   the source code into the format of the target environment specification (Col
   3:56-62 –This is achieved by the present invention by gathering
   information about the software application candidate for rehosting
   (including the source computing environment in which it operates),
   converting data files and related job control statements used on virtually
   any source host to the corresponding formats required by a new targeted
   host processing environment, Col 10:28-31 –Migration 400 automatically
   converts all files and data elements from the formats used on source

operating system 110 to the format requirements of target operating

system 450), *the custom knowledge base is updated to include additional*

structured information of the source platform and the source application

with respect to the target platform and the target environment specification

after each iteration to cause the knowledge engine to enhance source code

conversion in subsequent iterations (Col 2:66-67, Col 3:1-5 - The

discovery utility (knowledge Engine) is also suited to <u>analyze the</u>

<u>predetermined information to assess the feasibility of rehosting the</u>

<u>candidate software application</u> from the source computing environment

to the target computing environment. <u>A relational database management</u>

<u>system receives and stores the predetermined information and the</u>

<u>results of the analysis performed by the discovery utility</u> (Thus, relational

database saves information for next update on source application

according to the information stored and updated in the relational

database) –emphasis added); and

generating a report comprising a portion of the source code of the source

    application that is not converted automatically for manual conversion (Col

    3:35-40 -The system takes record descriptors (generation of reports) of

    data in the source environment that needs to be migrated and generates

    conversion programs for the data that get compiled. The compiled

    programs are then executed against the source data, producing output

    files that can be ported to the target environment; Col 9:43-47 –Based on

    the reports generated, a recommendation may be made by the present

    invention that the data be kept on a mainframe as a data repository

    while the application itself be ported to a mid-range environment; Col

    9:64-67 –A further illustration might be a case where the present

    invention "discovers" that the programs in the application to be rehosted

    consist of a mix of program languages, some of which cannot reasonably

    be converted by any means other than re-writing);

Pastilha sufficiently discloses a system and method for automatically re-
hosting (migrating) software system from a source computing environment to
target computing environment, wherein the said system and method provide an
utility component to assess both the source and target computing environment
based on the information gathered from a relational knowledge base regarding
said source and target environment, extracted from application software and
predetermined pieces of data, and further analyze the assessment, and
generates a description about both source and target computing environments.
A migration utility propagate re-hosting or migration process according to the
generated assessment and description (—See Pastilha Abstract, and Col 2-6),
but Pastilha does not explicitly discloses— obtaining User Interface (UI) details
of the source application receiving a validation scheme of a source front-end
interface, wherein the source code is passed through a knowledge engine for a
plurality of iterations;

however, Moore discloses—

obtaining User Interface (UI) details of the source application receiving a

> *validation scheme of a source front-end interface* (Refer to FIG. 4, 168,

> and associated text);

Thus it would have been obvious to one ordinary skill in the art at time the
invention was made to incorporate Moore's technique into Pastilha's invention
because the combined inventive system and method would provide an enhance
capability to automatically generate abstract intermediate representation of UI
information (front end), wherein the abstract representation would be used to
generate corresponding components in a new application shell in target
language or architecture (See Moore Paragraph [0039]), but Pastilha modified
by Moore does disclose: a plurality of iterations;

however, Lake discloses—

wherein the source code is passed through a knowledge engine for a plurality of

> *iterations* (Paragraph [0064] -The general methods of FIGS. 5A and 5B
> may be applied over multiple iterations in order to <u>fully update a
> database</u>, each <u>iteration using a different implementation</u> of the methods
> directed at different fields of the databases).

It would have been obvious to one ordinary skill in art at the time the invention
was made to use Lake's teaching in Pastilha modified by Moore's invention
because this inventive technique would enhance the discovery utility of
Pastilha to import the source and target environment information and extract
attributes from subjected application in order to implement the migration
process, and each iteration of the discovery utility may analyze attributes that
were generated through an earlier iteration, and update the database with the
new attributes (—See Lake Paragraph [0064]).

Per claim 14:

Pastilha discloses

> *dynamically hatch new patterns to be used to convert the source code into
> the format of the target environment specification* (Col 14:61-65 - if the source
> operating system from which the software is migrated does not require iterative
> file set handling, then the present <u>invention will create an equivalent of such
> iterative file set(s) for improved functionality</u> once the migrated software has
> been rehosted).

Per claim 17:

Pastilha discloses

> *wherein an existing knowledge base is provided for understanding the
> source application, the source platform, the target environment specification, and
> the target platform* (Col 2:61-67, Col 3:1-3 - A computer processor having a
> discovery utility is used to request predetermined information from the source
> computing environment including the operating system, any databases in the

source computing environment, as well as from the candidate software
application. The discovery utility is also suited to analyze the predetermined
information to assess the feasibility of rehosting the candidate software
application from the source computing environment to the target computing
environment).

Per claim 18:

Pastilha discloses

extracting the business logic and database schema of the source application
        systematically and logically (Col 5:43-50 -Discovery portion 140 of the
        present invention comprises requesting portion 145 which requests
        certain information from source operating system 110, <u>any database(s)</u>
        <u>125</u> (knowledge base) and application software 130. Requesting portion
        145 provides the results of its inquiries to data selector 150 for
        <u>extraction of certain predetermined pieces of data therefrom</u>. These
        pieces of <u>data are stored in relational database</u> management system
        ("RDBMS") 155; Col 134-38 –Rehosting is the physical movement of one
        or more business applications from one computing environment, such as
        mainframe, to another computing environment, such as a mid-range
        computing platform); and

converting the extracted business logic and database schema of the source
        application into a format specified for the target application (Col 5:43-50 -
        Discovery portion 140 of the present invention comprises requesting
        portion 145 which requests certain information from source operating
        system 110, <u>any database(s) 125</u> (knowledge base) and application
        software 130. Requesting portion 145 provides the results of its inquiries
        to data selector 150 for <u>extraction of certain predetermined pieces of data</u>
        <u>therefrom</u>. These pieces of <u>data are stored in relational database</u>
        management system ("RDBMS") 155; Col 134-38 –Rehosting is the

physical movement of one or more business applications from one
computing environment, such as mainframe, to another computing
environment, such as a mid-range computing platform).

9.   Claims 5 and 11 are rejected under 35 U.S.C. 103(a) as being
unpatentable over Pastilha in view of Moore and Lake, and further in view of
Uchikawa et al. (US Pat. No. 6,330,553 B1 –herein after Uchikawa) .

Per claim 5:
Pastilha modified by Moore and Lake does not disclose: wherein the processing
means is further configured to apply fuzzy rules to perform optimal conversion
of the source code into the format of the target environment specification;
however Uchikawa discloses—
 apply fuzzy rules to perform optimal conversion of the source code into the
format of the target environment specification (Col 2:45-54 –controlling an
object using at least one model which is obtained by modeling at least one part
of the internal structures of the object, wherein the modeling is conducted
using at least a fuzzy neural network which obtains appropriate numbers of
fuzzy rules in an autonomic manner by adding and deleting fuzzy rules. This
system allows easily and effectively controlling an object having high non-
linearity since fuzzy rules can be obtained simply and efficiently, thereby
efficiently modeling the object).
It would have been obvious to one ordinary skill in the art at time the invention
was made to include Uchikawa's teaching into Pastilha modified by Moore and
Lake's invention because the further incorporated technique would enhance
the modified invention a capability to use fuzzy neural network for converting
source platform information in a optimized way because the fuzzy inference
system allows linguistically descriptive algorithms including obscurity, such as
decision by humans, using if-then type fuzzy rules. The neural network allows

regulating any input-output relationship by updating coupling coefficients using a learning function –once suggested by Uchikawa (—See Uchikawa Col 1:7-52).

Per claim 6:

Pastilha modified by Moore and Lake does not disclose: wherein the processing means is further configured to utilize neural networks to convert the source code into the format of the target environment specification;

however Uchikawa discloses—

utilize neural networks to convert the source code into the format of the target environment specification (Col 2:45-54 –controlling an object using at least one model which is obtained by modeling at least one part of the internal structures of the object, wherein the modeling is conducted <u>using at least a fuzzy neural network which obtains appropriate numbers of fuzzy rules</u> in an autonomic manner by adding and deleting fuzzy rules. This system allows easily and effectively controlling an object having high non-linearity since fuzzy rules can be obtained simply and efficiently, thereby efficiently modeling the object).

It would have been obvious to one ordinary skill in the art at time the invention was made to include Uchikawa's teaching into Pastilha modified by Moore and Lake's invention because the further incorporated technique would enhance the modified invention a capability to use fuzzy neural network for converting source platform information in a optimized way because the fuzzy inference system allows linguistically descriptive algorithms including obscurity, such as decision by humans, using if-then type fuzzy rules. The neural network allows regulating any input-output relationship by updating coupling coefficients using a learning function –once suggested by Uchikawa (—See Uchikawa Col 1:7-52).

Per claim 9:

Pastilha modified by Moore and Lake does not disclose: applying fuzzy rules to perform optimal conversion of the source code into the format of the target environment specification;

however Uchikawa discloses—

apply fuzzy rules to perform optimal conversion of the source code into the format of the target environment specification (Col 2:45-54 –controlling an object using at least one model which is obtained by modeling at least one part of the internal structures of the object, wherein the modeling is conducted using at least a fuzzy neural network which obtains appropriate numbers of fuzzy rules in an autonomic manner by adding and deleting fuzzy rules. This system allows easily and effectively controlling an object having high non-linearity since fuzzy rules can be obtained simply and efficiently, thereby efficiently modeling the object).

It would have been obvious to one ordinary skill in the art at time the invention was made to include Uchikawa's teaching into Pastilha modified by Moore and Lake's invention because the further incorporated technique would enhance the modified invention a capability to use fuzzy neural network for converting source platform information in a optimized way because the fuzzy inference system allows linguistically descriptive algorithms including obscurity, such as decision by humans, using if-then type fuzzy rules. The neural network allows regulating any input-output relationship by updating coupling coefficients using a learning function –once suggested by Uchikawa (—See Uchikawa Col 1:7-52).

Per claim 10:

Pastilha modified by Moore and Lake does not disclose: utilizing neural networks to convert the source code into the format of the target environment specification;

however Uchikawa discloses—

utilize neural networks to convert the source code into the format of the target environment specification (Col 2:45-54 –controlling an object using at least one model which is obtained by modeling at least one part of the internal structures of the object, wherein the modeling is conducted <u>using at least a fuzzy neural network which obtains appropriate numbers of fuzzy rules</u> in an autonomic manner by adding and deleting fuzzy rules. This system allows easily and effectively controlling an object having high non-linearity since fuzzy rules can be obtained simply and efficiently, thereby efficiently modeling the object).

It would have been obvious to one ordinary skill in the art at time the invention was made to include Uchikawa's teaching into Pastilha modified by Moore and Lake's invention because the further incorporated technique would enhance the modified invention a capability to use fuzzy neural network for converting source platform information in a optimized way because the fuzzy inference system allows linguistically descriptive algorithms including obscurity, such as decision by humans, using if-then type fuzzy rules. The neural network allows regulating any input-output relationship by updating coupling coefficients using a learning function –once suggested by Uchikawa (—See Uchikawa Col 1:7-52).

Per claim 15:

Pastilha modified by Moore and Lake does not disclose: applying fuzzy rules to perform optimal conversion of the source code into the format of the target environment specification;

however Uchikawa discloses—

apply fuzzy rules to perform optimal conversion of the source code into the format of the target environment specification (Col 2:45-54 –controlling an object using at least one model which is obtained by modeling at least one part of the internal structures of the object, wherein the modeling is conducted <u>using at least a fuzzy neural network which obtains appropriate numbers of fuzzy rules</u> in an autonomic manner by adding and deleting fuzzy rules. This system allows easily and effectively controlling an object having high non-linearity since fuzzy rules can be obtained simply and efficiently, thereby efficiently modeling the object).

It would have been obvious to one ordinary skill in the art at time the invention was made to include Uchikawa's teaching into Pastilha modified by Moore and Lake's invention because the further incorporated technique would enhance the modified invention a capability to use fuzzy neural network for converting source platform information in a optimized way because the fuzzy inference system allows linguistically descriptive algorithms including obscurity, such as decision by humans, using if-then type fuzzy rules. The neural network allows regulating any input-output relationship by updating coupling coefficients using a learning function –once suggested by Uchikawa (—See Uchikawa Col 1:7-52).

Per claim 16:

Pastilha modified by Moore and Lake does not disclose: utilizing neural networks to convert the source code into the format of the target environment specification;

however Uchikawa discloses—

utilize neural networks to convert the source code into the format of the target environment specification (Col 2:45-54 –controlling an object using at least one model which is obtained by modeling at least one part of the internal structures of the object, wherein the modeling is conducted <u>using at least a</u>

fuzzy neural network which obtains appropriate numbers of fuzzy rules in an
autonomic manner by adding and deleting fuzzy rules. This system allows
easily and effectively controlling an object having high non-linearity since fuzzy
rules can be obtained simply and efficiently, thereby efficiently modeling the
object).

It would have been obvious to one ordinary skill in the art at time the invention
was made to include Uchikawa's teaching into Pastilha modified by Moore and
Lake's invention because the further incorporated technique would enhance
the modified invention a capability to use fuzzy neural network for converting
source platform information in a optimized way because the fuzzy inference
system allows linguistically descriptive algorithms including obscurity, such as
decision by humans, using if-then type fuzzy rules. The neural network allows
regulating any input-output relationship by updating coupling coefficients
using a learning function –once suggested by Uchikawa (—See Uchikawa Col
1:7-52).

Conclusion

10.   Applicant's amendment necessitated the new ground(s) of rejection
presented in this Office action. Accordingly, **THIS ACTION IS MADE FINAL**.
See MPEP § 706.07(a). Applicant is reminded of the extension of time policy as
set forth in 37 CFR 1.136(a).

A shortened statutory period for reply to this final action is set to expire THREE
MONTHS from the mailing date of this action. In the event a first reply is filed
within TWO MONTHS of the mailing date of this final action and the advisory
action is not mailed until after the end of the THREE-MONTH shortened
statutory period, then the shortened statutory period will expire on the date the
advisory action is mailed, and any extension fee pursuant to 37 CFR 1.136(a)

will be calculated from the mailing date of the advisory action. In no event, however, will the statutory period for reply expire later than SIX MONTHS from the date of this final action.

11.   Any inquiry concerning this communication or earlier communications from the examiner should be directed to ZIAUL CHOWDHURY whose telephone number is (571)270-7750. The examiner can normally be reached on Monday Thru Friday, 7:30AM To 9:00PM, Alternet Friday, Eastern Time.

If attempts to reach the examiner by telephone are unsuccessful, the examiner's supervisor, TUAN Q. DAM can be reached on 571-272-3695. The fax phone number for the organization where this application or proceeding is assigned is 571-273-8300.

Information regarding the status of an application may be obtained from the Patent Application Information Retrieval (PAIR) system. Status information for published applications may be obtained from either Private PAIR or Public PAIR. Status information for unpublished applications is available through Private PAIR only. For more information about the PAIR system, see http://pair-direct.uspto.gov. Should you have questions on access to the Private PAIR system, contact the Electronic Business Center (EBC) at 866-217-9197 (toll-free). If you would like assistance from a USPTO Customer Service Representative or access to the automated information system, call 800-786-9199 (IN USA OR CANADA) or 571-272-1000.

 /ZIAUL CHOWDHURY/   /Tuan Q. Dam/
Examiner, Art Unit 2192   Supervisory Patent Examiner, Art Unit 2192